

Claims

We claim:

Sub A/B)

1. A computer-implemented method for programmatically generating a graphical program based on a state diagram, comprising:
 - receiving state diagram information, wherein the state diagram information specifies one or more states;
 - programmatically generating the graphical program in response to the state diagram information, wherein said programmatically generating comprises
- 10 programmatically generating graphical source code corresponding to the specified one or more states.
2. The method of claim 1,
wherein the state diagram information represents a state diagram.
- 15 3. The method of claim 2,
wherein the state diagram represents desired operation of a software program.
- 20 4. The method of claim 2,
wherein the state diagram represents desired operation of a hardware device.
- 25 5. The method of claim 2,
wherein the state diagram represents a desired algorithm.
6. The method of claim 2,
wherein the state diagram represents a test sequence.
7. The method of claim 1,

wherein said programmatically generating the new graphical program creates the new graphical program without any user input specifying the new graphical program during said creating.

5 8. The method of claim 1,

wherein said programmatically generating the graphical program comprises programmatically generating a block diagram including the graphical source code corresponding to the specified one or more states.

10 9. The method of claim 1,

wherein the programmatically generated graphical source code includes placeholder graphical source code for each state.

15 10. The method of claim 9, further comprising:

for each state, a user manually entering graphical source code specifying execution instructions to be performed when the state is active during execution of the graphical program.

20 11. The method of claim 9,

wherein the placeholder graphical source code for each state comprises a case in a graphical case structure.

25 12. The method of claim 1,

wherein, for at least one state, the state diagram information specifies program code associated with the state;

wherein the programmatically generated graphical source code includes the specified program code.

13. The method of claim 1,

wherein, for at least one state, the state diagram information specifies program code associated with the state;

wherein the programmatically generated graphical source code is operable to invoke the specified source code.

5

14. The method of claim 1,

wherein the state diagram information further specifies one or more state transitions, wherein each state transition specifies a transition from a first state to a second state;

10 wherein said programmatically generating further comprises programmatically generating graphical source code corresponding to the specified state transitions.

15. The method of claim 14,

wherein the programmatically generated graphical source code includes 15 placeholder graphical source code for each state transition.

16. The method of claim 15, further comprising:

for one or more state transitions, a user manually entering graphical source code specifying a Boolean condition associated with the state transition.

20

17. The method of claim 14,

wherein the state diagram information specifies at least two state transitions from a particular state;

wherein the state diagram information also specifies a priority ordering for the at 25 least two state transitions;

wherein said programmatically generating comprises programmatically generating graphical source code such that, during execution of the graphical program, Boolean conditions associated with the at least two state transitions are evaluated in the specified priority ordering.

18. The method of claim 1,
wherein the state diagram information further specifies an initially active state;
wherein said programmatically generating comprises programmatically generating
5 graphical source code such that the graphical program begins execution in the initially
active state.

19. The method of claim 1,
wherein the state diagram information further specifies one or more stop states;
10 wherein said programmatically generating comprises programmatically generating
graphical source code such that if during execution of the graphical program one of the
stop states becomes active, then the graphical program is caused to stop execution.

20. The method of claim 1, further comprising:
15 receiving information specifying a change to the state diagram information;
programmatically updating the graphical program to reflect the specified change.

21. The method of claim 1,
wherein said programmatically generating the graphical program comprises
20 calling an application programming interface (API) enabling the programmatic
generation of a graphical program.

22. The method of claim 1,
wherein said programmatically generating the graphical program comprises
25 programmatically requesting a server program to generate the graphical program.

23. A computer-implemented method for programmatically generating a new graphical program, comprising:

providing information specifying a state diagram;

executing a graphical program generation (GPG) program;

5 the GPG program programmatically generating the new graphical program using said information, wherein the new graphical program includes graphical source code corresponding to the state diagram.

24. The method of claim 23, wherein said programmatically generating the
10 new graphical program creates the new graphical program without any user input specifying the new graphical program during said creating.

15 25. A computer-implemented method for programmatically generating a graphical program based on a state diagram, comprising:

displaying an initial state diagram;

programmatically generating a graphical program corresponding to the initial state diagram;

receiving user input specifying a change to the initial state diagram;

20 programmatically updating the graphical program to correspond to the specified change, in response to the user input specifying the change.

25 26. A system for programmatically generating a graphical program, the system comprising:

a processor coupled to a memory, wherein the memory stores a graphical program generation (GPG) program;

wherein the processor is operable to execute the GPG program in order to:

receive state diagram information, wherein the state diagram information specifies one or more states;

5 programmatically generate the graphical program in response to the state diagram information, wherein said programmatically generating comprises programmatically generating graphical source code corresponding to the specified one or more states.

27. The system of claim 26,

10 wherein said programmatically generating the graphical program creates the graphical program without any user input specifying the graphical program during said creating.

28. The system of claim 26,

15 wherein said programmatically generating the graphical program comprises programmatically generating a block diagram including the graphical source code corresponding to the specified one or more states.

20 29. A memory medium for programmatically generating a graphical program based on a state diagram, the memory medium comprising program instructions executable to:

receive state diagram information, wherein the state diagram information specifies one or more states;

25 programmatically generate the graphical program in response to the state diagram information, wherein said programmatically generating comprises programmatically generating graphical source code corresponding to the specified one or more states.

30. The memory medium of claim 29,

wherein said programmatically generating the new graphical program creates the new graphical program without any user input specifying the new graphical program during said creating.

5 31. The memory medium of claim 29,

wherein said programmatically generating the graphical program comprises programmatically generating a block diagram including the graphical source code corresponding to the specified one or more states.

10

PRINTED IN U.S.A. ON RECYCLED PAPER BY RICOH CO., LTD.